17th ICCRTS

"Operationalizing C2 Agility"

Draft Paper 053 (Accepted)

Multi-INT Complex Event Processing using Approximate, Incremental Graph Pattern Search

Topics

Primary: Topic 3 - Data, Information and Knowledge

Name of Authors

Dr. Jim Law and Dr. Scott McGirr
jim.law@navy.mil, scott.mcgirr@navy.mil

Point of Contact

Dr. Jim Law
jim.law@navy.mil

Name of Organization

SPAWAR Systems Center Pacific
53560 Hull Street
San Diego, CA USA 92152
1-619-553-2449
jim.law@navy.mil

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
JUN 2012		**00-00-2012 to 00-00-2012**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Multi-INT Complex Event Processing using Approximate, Incremental Graph Pattern Search	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Space and Naval Warfare Systems Center,53560 Hull Street,San Diego,CA,92152-5001**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
Presented at the 17th International Command & Control Research & Technology Symposium (ICCRTS) held 19-21 June, 2012 in Fairfax, VA.

14. ABSTRACT
Complex Event Processing, or CEP, is an event processing technique that analyzes multiple events with the goal of identifying meaningful complex events within an event cloud. CEP employs techniques such as detection of complex patterns of many events, event correlation and abstraction, computable event hierarchies, and event relationships such as causality, membership, timing, and event sequences. In this paper we present a detailed analysis of the characteristics of Multi-INT data streams from an Expeditionary and Irregular Warfare (EIW) environment. After evaluating these characteristics we propose a solution using approximate, incremental graph pattern search algorithms. Finally, we present a prototype implementation of these algorithms and a preliminary evaluation of their use and performance.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**30**	

Multi-INT Complex Event Processing using Approximate, Incremental Graph Pattern Search

James B. Law, Ph.D. and Scott C. McGirr, Ph.D.
Space and Naval Warfare Systems Center Pacific (SSC Pacific)
53560 Hull Street, San Diego, CA 92152-5001, USA
jim.law@navy.mil, scott.mcgirr@navy.mil

Abstract:

Complex Event Processing, or CEP, is an event processing technique that analyzes multiple events with the goal of identifying meaningful complex events within an event cloud. CEP employs techniques such as detection of complex patterns of many events, event correlation and abstraction, computable event hierarchies, and event relationships such as causality, membership, timing, and event sequences. In this paper we present a detailed analysis of the characteristics of Multi-INT data streams from an Expeditionary and Irregular Warfare (EIW) environment. After evaluating these characteristics we propose a solution using approximate, incremental graph pattern search algorithms. Finally, we present a prototype implementation of these algorithms and a preliminary evaluation of their use and performance.

Keywords: multi-INT, data fusion, complex event processing, graph theory, graph patterns, graph search, expeditionary warfare.

1. Introduction.

A complex event is a composite of simpler events. The components of a particular complex event are frequently variable and spread across a significant period of time. Complex Event Processing (CEP) is concerned with finding these complex events in both large collections of events and event streams. Many intelligence gathering systems produce high volume input and output streams of simple events [1]. Many systems also store events for some period of time depending on user need for history, information fusion, or other system processing. Warfighters demand that these streams and collections be examined often and in near real-time for situation awareness, force protection, and force projection. To meet this demand, processing strategies and algorithms are needed to automate detection of events in clutter.

The literature of ideas in CEP appear to have had their genesis in the active database community [3, 47], and discussion continues recently [41]. Current CEP techniques have been widely

discussed in the business-oriented data management community for a lengthy period [9, 19, 20, 21, 23, 25, 32, 41, 43, 46]. Additional, application areas include intrusion detection [17], provenance and workflow management [40], and software maintenance [28].

Our first approach to detecting complex events in multi-INT Expeditionary and Irregular Warfare (EIW) data streams was to try adapting current business-oriented CEP techniques for use on multi-INT data streams. In the next subsection of this paper we will present our analysis difficulties and shortcomings of this initial approach. In the second section of this paper we will discuss an approach that we took and the expected benefits. In the third section we present our dynamic complex event processing algorithms based on graph pattern search. In the fourth section we will outline our prototype implementation and show an example of its operation. In the fifth and final section we will present the results of a preliminary examination of the performance of our algorithms compared to a standard search algorithm.

2. Multi-INT data streams.

Our initial examination of current event processing techniques and implementations revealed that the data operated on is generally well defined. For example, financial transactions are usually generated by machines and while the volume may be high, the definitions and character of the messages is rigidly defined and displays little variance. In contrast, military messages are frequently generated by humans, and while the format of the messages may be defined, the content is often not constrained. Reporting is often subjective, arbitrarily delayed, and incomplete. This makes current open-source and commercial CEP engines [12, 33, 34, 37, 38, 45] unsuitable for use in Expeditionary and Irregular Warfare (EIW) environments. This presented a challenge that we undertook by characterization of event streams, generalized search methods and use of dynamic algorithms best suited for EIW user needs.

Since our work is aligned with the EIW development environment the first step we undertook was an informal survey of related projects to determine data storage facilities and formats. We found that graph-based notations are dominant in EIW Science and Technology (S&T) projects. Projects are increasingly sharing results of data analysis in Resource Description Format (RDF) [30], a recently released World Wide Web Consortium (W3C) [44] standard for flexible knowledge markup. Figure 1 show a typical workflow of several US Navy development projects we reviewed. RDF is an explicitly specified directed graph format where nodes represent known entities and directed edges represent a defined relationship between the source and sink nodes. The use of RDF enables data and service sharing but for extended use requires definitions of entities and relationships and this is still in flux. Despite this uncertainty we feel that graphical notations will dominate future data storage and processing environments in EIW.

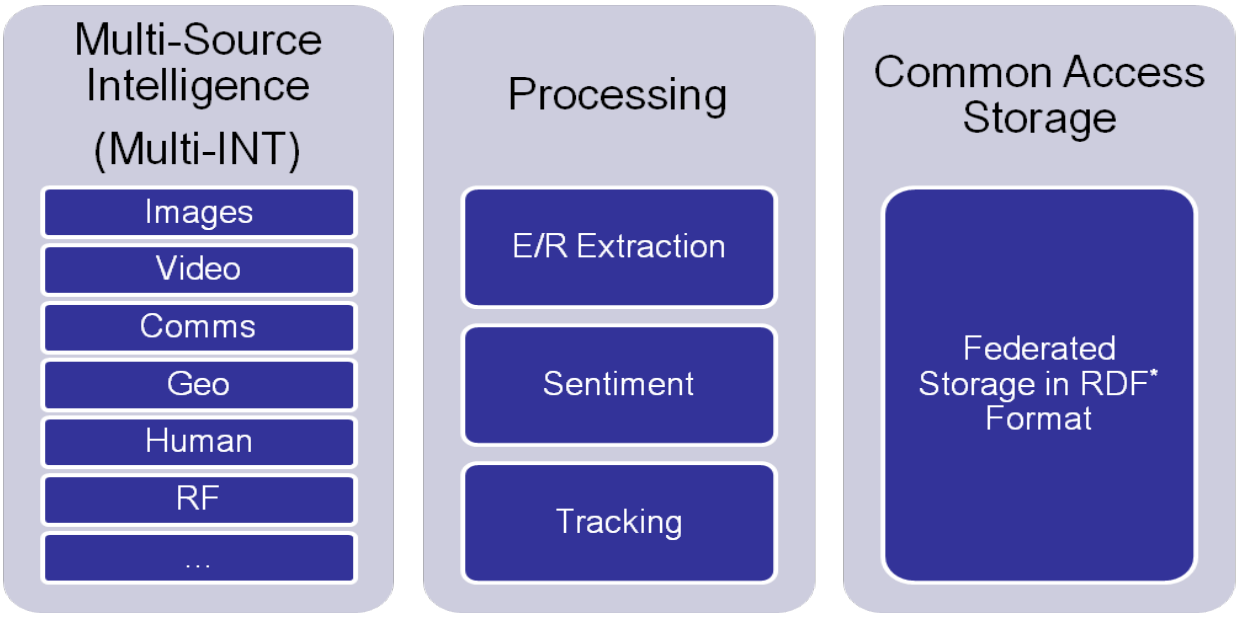


Figure 1. The Expeditionary and Irregular Warfare (EIW) data environment requires multi-INT data sources. The S&T community is increasingly using RDF for data storage.

The essential function in any event processing system is some method of pattern finding. The RDF standard has an associated query language named SPARQL [36]. SPARQL on an RDF store serves a purpose similar to SQL on a relational database store. One problem in using SPARQL to find complex events is that the presentation of events is variable. Significant event patterns can be buried in large amounts of data over extended time periods and thus not evident or require multiple queries. Additionally, the RDF store undergoes considerable change and may contain gaps in information or out of order entry of information. Due to these factors, the results of a query can be non-deterministic. Since the essential format of RDF information is a graphical network we conducted a survey of network algorithms available in the literature.

For our survey we divided the network literature into three distinct regimes: static networks [35], dynamic networks [4, 11, 14, 29], and event sequences [2, 5, 7, 8, 15, 16, 24, 26, 47]. Figure 2 shows the general characteristics of each of these literature regimes. In general we find that EIW data streams contain very little static information. Static information is limited to physical geography and major structures. While social networks are usually present, the relations are not predominately concurrent and relations and dependencies across the data are complex. Similarly, while much of the data consists of some approximate sequence, sequence may be only roughly decidable. Likewise, persistence and concurrency are variable. In contrast, characterizations of dynamic network data fit the EIW data environment well for reasons shown in the figure.

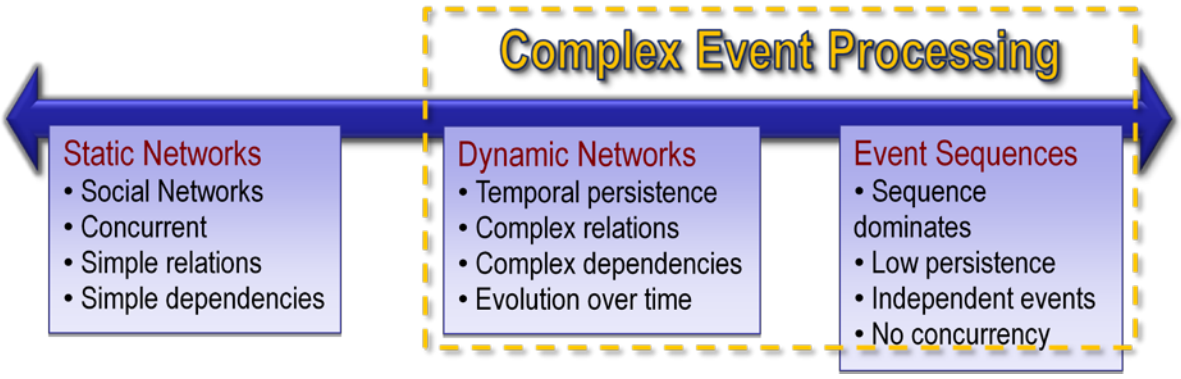


Figure 2. Three distinct network regimes have been explored in the literature. CEP in EIW event streams rarely has the characteristics of static network data.

3. Approximate, incremental graph pattern search.

The characteristics of C4ISR data environments, and EIW in particular, suggest a more general approach than those previously reported in CEP literature. Since the information environment is often stored and visualized as a graph this suggests an approach based on some method of approximate graph pattern matching [6, 10, 13, 29, 39, 49]. However, the data stream creates a dynamic environment. In this section we will present such an algorithm based on new work [13] in combination with previous incremental update techniques for dynamic graphs [29]. Our requirements are depicted graphically in Figure 3. We wish to search for approximate matches across RDF defined by multiple ontologies and potentially disconnected graphs.

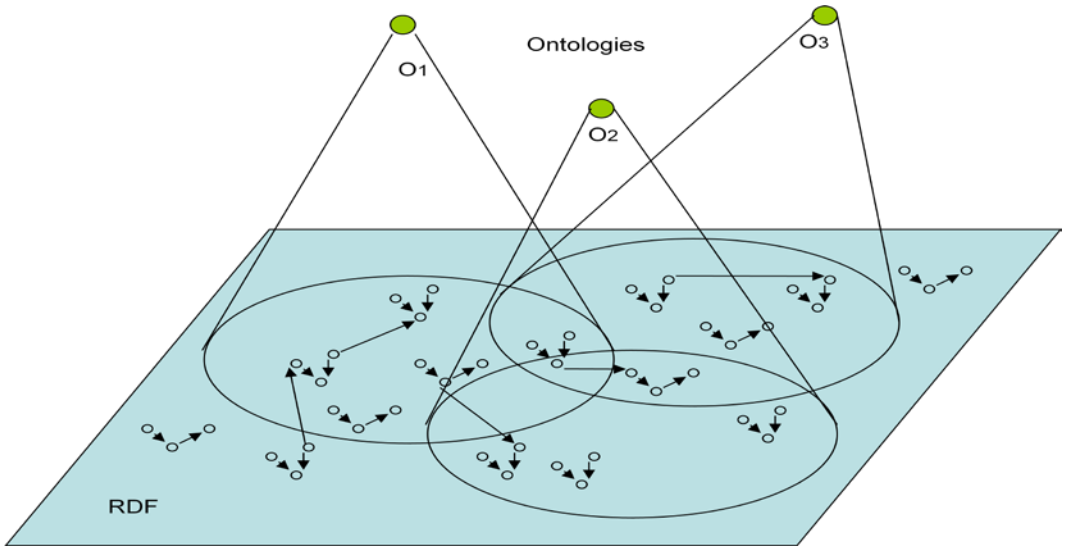


Figure 3. The graphic illustrates how data triples (subject-predicate-object) can be associated into ontologies (O1, O2, O3) and these in turn can be mapped into a complex graph.

Of particular interest to our research, was a polynomial time algorithm for graph pattern search and constructs a proof of its run-time [13]. This suggests that their algorithm will scale well enough to handle large data sets encountered in many C4ISR applications. However, since it assumes a static graph, it does not take into account the scale of change in the underlying graph that would be expected. Earlier work by Ramalingam [29] provides a basis for incrementally updating data structures that may allow tractable graph pattern searches. Figure 4 shows pseudo code for our graph pattern match algorithm [29].

Graph Pattern Match

Input: Pattern P=(Vp,Ep), Data Graph G=(V,E), and Ontology O

Initial step: compute all-pairs-shortest path matrix.

A. Find descendent nodes, D_{Vp}=desc(Vp), in ontology of each pattern graph node.

B. Compute potential matching set in G for each node in D_{Vp}.

C. Traverse paths in the matching set, examining path length and edge type. Remove nodes that are not connected, do not meet path constraints, or do not have correct edge type.

Figure 4. Pseudocode for our graph pattern match algorithm. The distance matrix M is updated separately using the algorithm in [29].

4. Prototype implementation.

In this section we describe out proposed system and prototype implementation of our algorithms, and present an example using data generated by the US Marine Corps. The architecture of our proposed system is shown in Figure 5. For our study, we assume to have access to a variety of multi-INT data streams in RDF format that could be stored and managed centrally. The lifetime of the data will be expected to vary with the capacity of the overall system and the needs of the processing systems generating and consuming the data. A prototype was constructed to show a proof-of-concept for identifying complex events. The complex events are built from simple events that can arrive through separate event streams. It is necessary to combine the data through a common lexicon and ontology. Python scripts were used to simplify prototype implementation.

The data used for the proof-of-concept was from a Second Marine Expeditionary Force (IIMEF) experiment that took place a Camp Jejune, Dec 13-15, 2011. A use case was constructed for emplacing an Improvised Explosive Device (IED). This involved vehicles, individual dismounts,

and activity alongside a road. There was contextual information and prior relationships established of vehicles, individuals and area that activity occurred. The data arrived in the form of Intel reports (e.g. DIIRs) and tactical reports (e.g. TACREPS). This data was tagged, associated and analyzed. The collection consists of 35 short text reports prepared during the first phase of the exercise, Intelligence Preparation for the Battlefield (IPB). The IPB reports describe a fictional background scenario spanning several days before a Marine squad undertakes movement in to and then out of a fictional Afghan village. The objective of our CEP system is to identify potential IED-related activity in these reports.

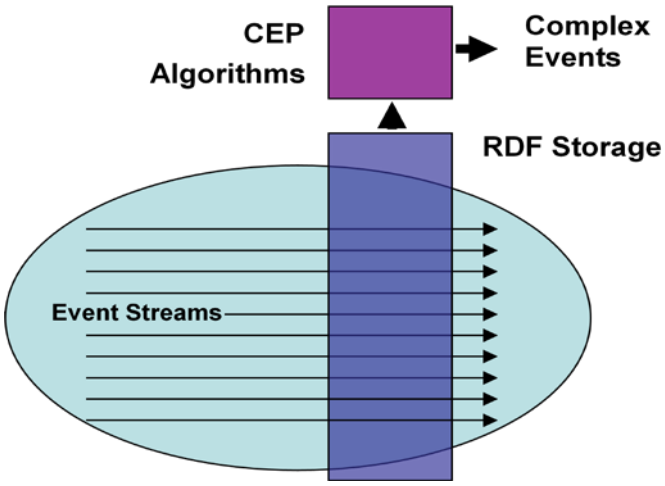


Figure 5. Graphic diagram of processing method that inputs event streams, converts data sources to metadata, analyzes metadata with CEP algorithms and outputs complex events.

We reduced the content of the reports to RDF by hand. In a working prototype this step would be automated in cooperation with other systems to reliably identify entities and relationships and encode them. The output of our encoding into RDF of all IPB reports. The complexity of the graph makes human interpretation very difficult. In addition, we created an informal ontology, for the objects and relationships in the IPB reports. This ontology is used to find the descendent nodes for a pattern matching algorithm. The hierarchy of entities and relationships can infer groups and similarities. For example, a storage facility may be any building or enclosed structure, and a vehicle may be a car, truck, or bus. In practice, an ontology would be developed cooperatively with the other systems contributing to the multi-INT data streams.

Finally, Figure 6 shows an example of a graph pattern specifying a potential threat related to IED activity. In this case we are looking for persons with a previous association with some type of IED activity who have direct access, or are linked to persons with access to, fertilizer, a vehicle, and a storage facility. For example, IED activity could include IED funding, manufacture, placement, transport, or triggering. Examples of a vehicle could include cars, trucks, and other vehicles. A storage facility may include a shop, house, or out building.

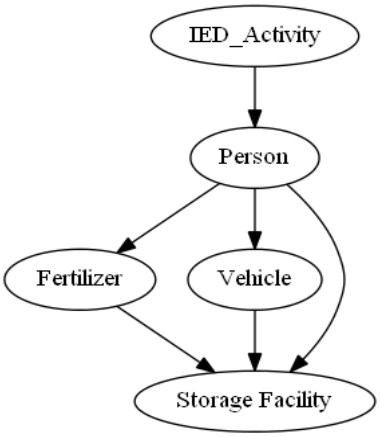


Figure 6. An example of a graph pattern for complex events related to IEDs. In this case we are searching for any set of relationships involving a person previously linked to IED activity and persons with access to fertilizer, a vehicle, and a storage facility.

The IPB reports which were the source of the colored nodes in the threat warning output are shown in Figure 7. The full data graph with similarly colored nodes is shown in Figure 8. The IPB reports were processed roughly in the order shown. It can be seen that earlier information in reports DIIR 1-05 and report DIIR 1-08 was later connected with information in TACREP 4-13. Our system builds data graph incrementally and raises a threat warning whenever the a match is found for the specified pattern graph. The pattern graph produced that is of interest is depicted in Figure 9.

Draft Intelligence Information Reports

DIIR_1-01.txt	DIIR_1-03.txt
DIIR_1-04.txt	DIIR_1-05.txt
DIIR_1-06.txt	DIIR_1-08.txt
DIIR_2-01.txt	DIIR_2-02.txt
DIIR_2-05.txt	DIIR_3-01.txt
DIIR_3-03.txt	DIIR_3-04.txt
DIIR_4-01.txt	DIIR_4-03.txt
DIIR_4-05.txt	DIIR_4-08.txt
DIIR_4-10.txt	DIIR_4-14.txt
DIIR_4-17.txt	

TACREPS

TACREP_1-02.txt	TACREP_2-03.txt
TACREP_2-04.txt	TACREP_2-06.txt
TACREP_3-02.txt	TACREP_4-02.txt
TACREP_4-04.txt	TACREP_4-06.txt
TACREP_4-09.txt	TACREP_4-11.txt
TACREP_4-12.txt	TACREP_4-13.txt
TACREP_4-16.txt	TACREP_4-18.txt
TACREP_4-19.txt	TACREP_4-21.txt

Figure 7. The raw data for our example is contained in two sets of reports. The color highlights show reports that are linked to IED activity.

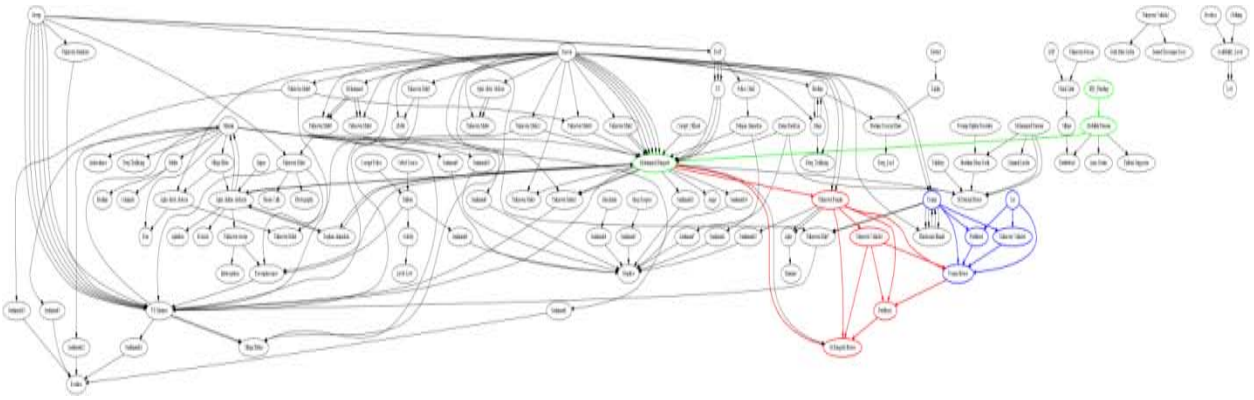


Figure 8. The full RDF data IED related activity.

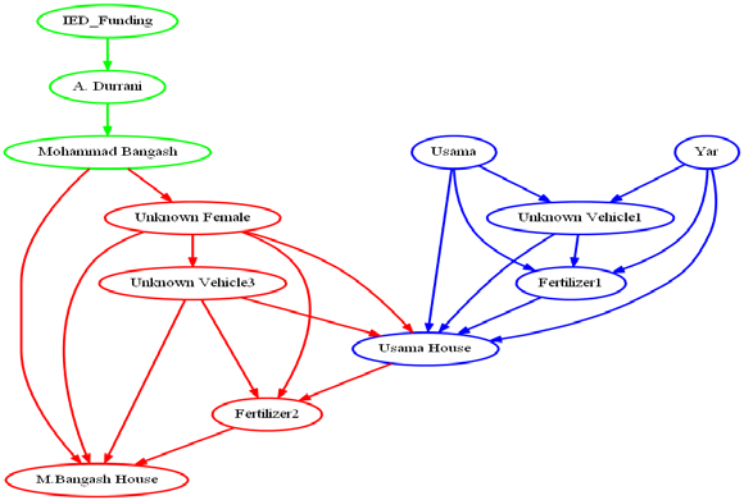


Figure 9. The threat warning graph generated by our graph pattern match algorithm.

5. Preliminary performance evaluation.

We tested out prototype against a well-developed library implementation of SPARQL. Figure 10 summarizes the results. We used synthetic data sets so that we could vary the size consistently and control the complexity of the RDF sets [18]. Our graph pattern algorithm prototype consisted of a 680 line implementation in Python using the networkx graph library [27] and the rdflib RDF library [31]. The SPARQL query prototype was a 200 line implementation in Python using the rdflib RDF library and the rdflib SPARQL library.

The results of our test show that the running times of the two implementations was significantly different at a confidence level of 95 percent. The difference in run time for the SPARQL implementation for the 10,000 and 100,000 RDF triple tests was not statistically significant at a 95 percent confidence level. The results are shown in Figure 10.

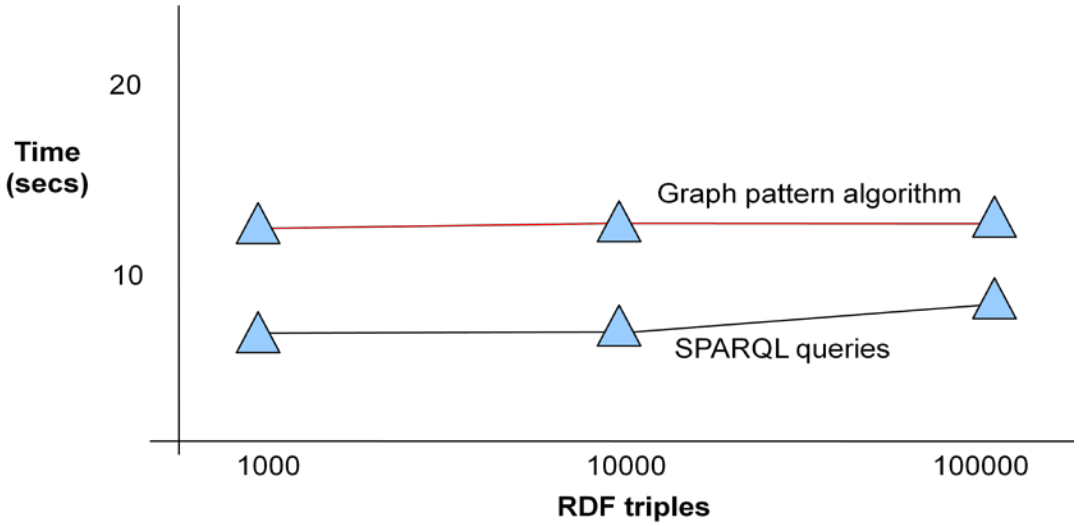


Figure 10. A comparison of graph pattern search and SPARQL queries. Total execution time for 10 executions each of 5 random pattern searches in synthetic data sets.

Statistics were gathered for the six test runs of the prototype graph pattern search implementation and SPARQL query standard algorithm. The execution environment was a Dell Precision T1500 desktop PC with Core i7 processor, 8GB of RAM, Windows 7 operating system, and a one TB hard drive. Fifty runs were made for each RDF triple graph. From these preliminary tests we conclude that our graph pattern search algorithms have a runtime performance that is acceptable at this early stage of investigation. We are currently engaged in implementing a more extensive prototype which we can use to test in more realistic data environments.

6. Summary.

In summary we have presented an analysis of the generalized EIW multi-INT data environment and an approximate graph pattern search algorithm for identifying complex events. We tested our prototype algorithms against standard search algorithms and determined that the performance is acceptable. In our view the preliminary performance of the system is adequate to justify further research investment. There is a wide range of potential EIW datasets.

Our future work will involve continued development of the prototype discussed in this paper. The objective of the next phase of research will be demonstrating operation in the streaming environment of a Marine Corps exercise. We will also seek to evaluate and better understand the developing data environment and adapt our algorithms as necessary.

Acknowledgements.

This research was funded by the Office of Naval Research (ONR). We appreciate their financial support, technical advice and assistance in obtaining data to evaluate event processing concepts. We are grateful to the technical library staff at SSC Pacific for assistance in obtaining references. This paper is the work of U.S. Government employees performed in the course of employment and no copyright subsists therein.

References.

1. AIR FORCE RESEARCH LABORATORY ID. SURVEY OF EVENT PROCESSING. *dtic.mil*. 2007;(December). Available at: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:SURVEY+OF+EVENT+PROCESSING#0.

2. Allan J, Papka R, Lavrenko V. On-line new event detection and tracking. In: *Proceedings of the 21st annual International ACM SIGIR conference on research and development in information retrieval*. ACM New York, NY, USA; 1998:37–45. Available at: http://portal.acm.org/citation.cfm?id=290941.290954&type=series.

3. Biossfeld H, Rohwer G. Techniques of Event History Modeling. *New Approaches to Causal Analysis. Mahwah und*. 1995. Available at: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Techniques+of+event+history+modeling. Accessed March 11, 2010.

4. Brandes U, Lerner J, TAB. Networks Evolving Step by Step: Statistical Analysis of Dyadic Event Data. *Network Analysis*. 2009:1-6. Available at: http://doi.ieeecomputersociety.org/10.1109/ASONAM.2009.28.

5. Brenna L, Demers A, Gehrke J, et al. Cayuga: a high-performance event processing engine. In: *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*. ACM; 2007:1102. Available at: http://portal.acm.org/citation.cfm?id=1247620.

6. Bröcheler M, Pugliese A. Probabilistic Subgraph Matching on Huge Social Networks. In: *International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2011*. Kaohsiung; 2011:271 - 278. Available at: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Probabilistic+Subgraph+Matching+on+Huge+Social+Networks#0. Accessed July 7, 2011.

7. Butts CT. A relational event framework for social action. *Sociological Methodology*. 2008;38(1):155–200. Available at: http://www3.interscience.wiley.com/journal/121358128/abstract.

8. Chakravarthy S, Krishnaprasad V, Anwar E, Kim SK. Composite events for active databases: Semantics, contexts and detection. In: *Proceedings of the International Conference on Very Large Data Bases*. Citeseer; 1994:606–606. Available at: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Composite+events+for+active+databases:+Semantics,+contexts+and+detection#0.

9. Chandrasekaran S, MJ, Cooper O, Deshpande A. TelegraphCQ: Continuous dataflow processing for an uncertain world. In: *Proc. of CIDR.*; 2003. Available at: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:TelegraphCQ:+Continuous+dataflow+processing+for+an+uncertain+world#1.

10. Chen L, Gupta A, Kurul ME. Stack-based algorithms for pattern matching on dags. In: *Proceedings of the 31st international conference on Very large data bases*. VLDB Endowment; 2005:493–504. Available at: http://portal.acm.org/citation.cfm?id=1083592.1083651. Accessed February 8, 2011.

11. Conway D. Modeling Network Evolution Using Graph Motifs. *ArXiv preprint :1105.0902v1*. 2011.

12. Esper. *http://esper.codehaus.org/*

13. Fan W, Li J, Ma S, Tang N, Wu Y. Graph Pattern Matching: From Intractable to Polynomial Time. *Proceedings of the VLDB.* 2010;3(1):264-275. Available at: http://www.vldb2010.org/proceedings/files/papers/R23.pdf. Accessed March 3, 2011.

14. Friedman N, Getoor L, Koller D, Pfeffer A. Learning probabilistic relational models. In: *International Joint Conference on Artificial Intelligence*.Vol 16. Citeseer; 1999:1300–1309. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.31.4737&rep=rep1&type=pdf.

15. Gyllstrom D, Agrawal J, Diao Y, Immerman N. On supporting kleene closure over event streams. In: *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*. Citeseer; 2008. Available at: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:On+Supporting+Kleene+Closure+over+Event+Streams#0.

16. Kalashnikov DV, Ma Y, Mehrotra S, Hariharan R, Butts C. Modeling and querying uncertain spatial information for situational awareness applications. In: *Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems*. ACM; 2006:138. Available at: http://portal.acm.org/citation.cfm?id=1183471.1183494.

17. Krügel C, Toth T, Kerer C. Decentralized event correlation for intrusion detection. *Information Security and Cryptology—ICISC*. 2001:1-21. Available at: http://www.springerlink.com/index/NWXHUWH7E5LCDWRU.pdf.

18. Law JB, Ceruti MG. Supporting C2 Research and Evaluation□: An Infrastructure and its Potential Impact Supporting C 2 Research and Evaluation□: An Infrastructure and its Potential Impact. In: *16th ICCRTS*.; 2011:21-23.

19. Luckham DC, Frasca B. Complex event processing in distributed systems. *Computer Systems Laboratory Technical Report CSL-TR-98-754. Stanford University, Stanford*. 1998:-98-754. Available at: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Complex+event+processing+in+distributed+systems#0.

20. Luckham DC, Vera J, Bryan D, Augustin L, Belz F. Partial orderings of event sets and their application to prototyping concurrent, timed systems. *Journal of Systems and Software*. 1993;21(3):253–265. Available at: http://linkinghub.elsevier.com/retrieve/pii/016412129390027U.

21. Luckham D. Is There a Commercial Need for a Quantum Leap in CEP Technology? 1 Part 2. *Event (London)*. 2008:1-6.

22. Luckham D. SOA , EDA , BPM and CEP are all Complementary. *Power*. 2007.

23. Martin-Flatin JP, Jakobson G, Lewis L. Event Correlation in Integrated Management: Lessons Learned and Outlook. *Journal of Network and Systems Management*. 2007;15(4):481-502. Available at: http://www.springerlink.com/index/10.1007/s10922-007-9078-5.

24. Mehrotra S, Butts C, Kalashnikov D. Project RESCUE: challenges in responding to the unexpected. *SPIE*. 2004;5304(January):179-192. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.77.6840&rep=rep1&type=pdf. Accessed July 20, 2011.

25. Mendes MRN, Bizarro P, Marques P. A framework for performance evaluation of complex event processing systems. *Proceedings of the second international conference on Distributed event-based systems - DEBS '08*. 2008:313. Available at: http://portal.acm.org/citation.cfm?doid=1385989.1386030.

26. Nallapati R, Feng A, Peng F, Allan J. Event threading within news topics. In: *Proceedings of the thirteenth ACM international conference on Information and knowledge management*. ACM; 2004:446–453. Available at: http://portal.acm.org/citation.cfm?id=1031258. Accessed March 10, 2010.

27. NetworkX. *http://networkx.lanl.gov/*.

28. Popescu D, Garcia J, Bierhoff K, Medvidovic N. Helios: Impact analysis for event-based systems. *ICSE 2010*. 2010. Available at: http://csse.usc.edu/csse/TECHRPTS/2009/usc-csse-2009-517/usc-csse-2009-517.pdf.

29. Ramalingam G, Reps T. An incremental algorithm for a generalization of the shortest-path problem. *Journal of Algorithms*. 1996;21(2):267–305. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.45.1855&rep=rep1&type=pdf. Accessed February 23, 2011.

30. RDF. World Wide Web Consortium (W3C). Resource Description Framework (RDF): Concepts and Abstract Syntax. *Latest version available at:* http://www.w3c.org/TR/rdf-concepts

31. RDFLIB. *https://github.com/RDFLib*. Documentation for rdflib can be found at: *http://readthedocs.org/docs/rdflib/*.

32. Rizvi S. *Complex Event Processing Beyond Active Databases: Streams and Uncertainties*. 2005. Available at: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Complex+Event+Processing+Beyond+Active+Databases:+Streams+and+Uncertainties#0.

33. S4. *http://s4.io/*

34. Siddhi. *http://siddhi.sourceforge.net/*

35. Snijders TAB. Models for longitudinal network data. *Models and methods in social network analysis*. 2005:215–247. Available at: http://books.google.com/books?hl=en&lr=&id=4Ty5xP_KcpAC&oi=fnd&pg=PA215&dq=Models+for+longitudinal+network+data&ots=9KJPvfxbF0&sig=o4vFlsQD5DUO7YOPkBSNFsxM4n8.

36. SPARQL. W3C. SPARQL Query Language for RDF. Available at: http://www.w3.org/TR/rdf-sparql-query/. Accessed March 6, 2012.

37. StreamBase. *http://www.streambase.com/sbx.htm*

38. TIBCO. *http://www.tibco.com/products/business-optimization/complex-event-processing/default.jsp*

39. Tong H, Faloutsos C, Gallagher B, Eliassi-Rad T. Fast best-effort pattern matching in large attributed graphs. *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '07*. 2007:737. Available at: http://portal.acm.org/citation.cfm?doid=1281192.1281271.

40. Vijayakumar NN, Plale B. Tracking Stream Provenance in Complex Event Processing Systems for Workflow-Driven Computing. In: *VLDB*. Vienna, Austria: ACM New York, NY,

USA; 2007. Available at: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Tracking+Stream+Provenance+in+Complex+Event+Processing+Systems+for+Workflow-Driven+Computing#0.

41. Wasserkrug S, Gal A, Etzion O, Turchin Y. Complex event processing over uncertain data. In: *Proceedings of the second international conference on Distributed event-based systems*. ACM; 2008:253–264. Available at: http://portal.acm.org/citation.cfm?id=1386022.

42. White S. Event Server: A Lightweight, Modular Application Server for Event Processing Seth. *blogs.oracle.com*. 2008:193-200. Available at: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:WebLogic+Event+Server:+A+Lightweight,+Modular+Application+Server+for+Event+Processing+Seth+White#1.

43. Widder A, Ammon RV, Schaeffer P, Wolff C. Identification of suspicious, unknown event patterns in an event cloud. In: *Proceedings of the 2007 inaugural international conference on Distributed event-based systems - DEBS '07*. New York, New York, USA: ACM Press; 2007:164. Available at: http://portal.acm.org/citation.cfm?doid=1266894.1266926.

44. World Wide Web Consortium (W3C). RDF Primer. Available at: http://www.w3.org/TR/2004/REC-rdf-primer-20040210/. Accessed March 6, 2012.

45. WSO2. *http://wso2.com/products/complex-event-processing-server/*

46. Wu E, Diao Y, Rizvi S. High-performance complex event processing over streams. In: *Proceedings of the 2006 ACM SIGMOD international conference on Management of data*.Vol pages. ACM; 2006:418. Available at: http://portal.acm.org/citation.cfm?id=1142473.1142520.

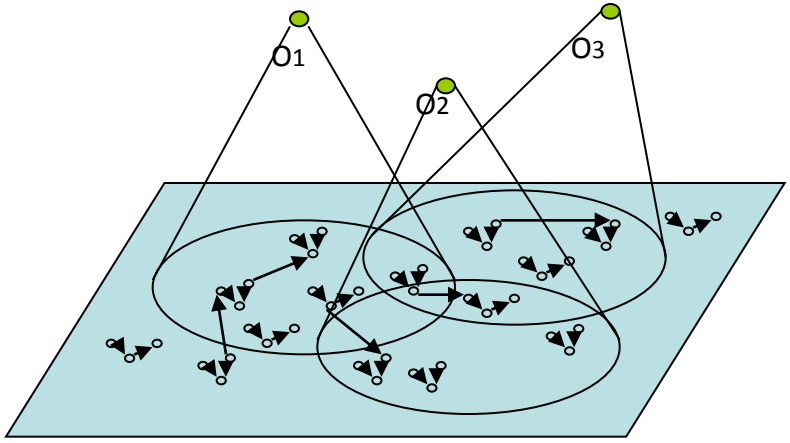
47. Yamaguchi K. Event-history analysis: it's contributions to modeling and causal inference. *Sociological Theory and Methods*. 1987;2(1):61-82.

48. Zimmer D, Unland R. On the semantics of complex events in active database management systems. In: *Proceedings of the international conference on data engineering*. INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS; 1999:392–399. Available at: http://doi.ieeecomputersociety.org/10.110910.1109/ICDE.1999.754955.

49. Zou L, Chen L, Ozsu MT. Distance-join: Pattern match query in a large graph database. *Proceedings of the VLDB Endowment*. 2009;2(1):886–897. Available at: http://portal.acm.org/citation.cfm?id=1687727. Accessed February 23, 2011.

Multi-INT Complex Event Processing using Approximate, Incremental Graph Pattern Search

Dr. Jim Law and Dr. Scott McGirr
SPAWAR Systems Center Pacific
San Diego, CA, USA

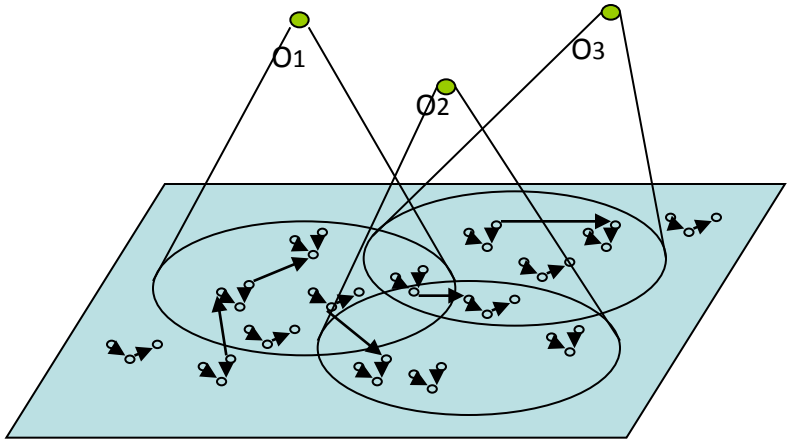


POC: Jim Law
619-553-2449
jim.law@navy.mil

Technical Approach

Problem: Dynamic Data

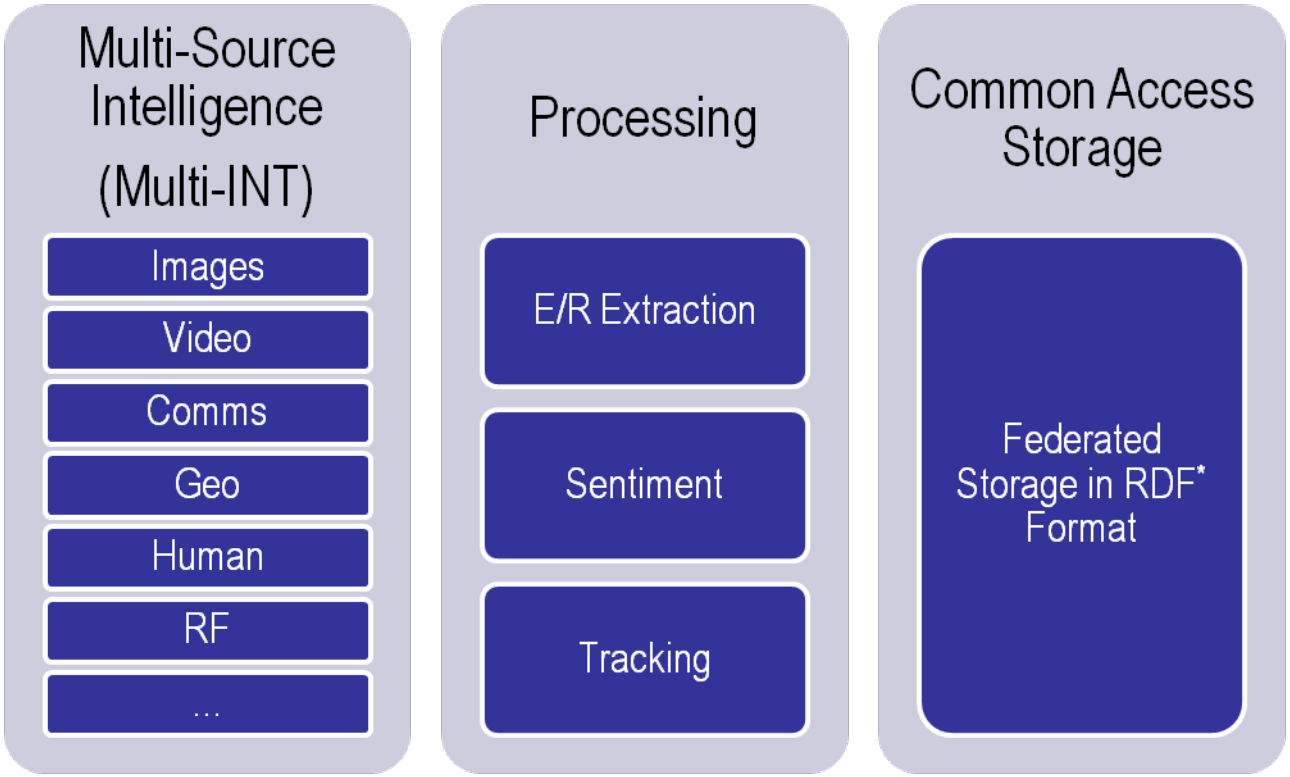
Constantly changing data from many sources leads to fragmented, inefficient search methods.

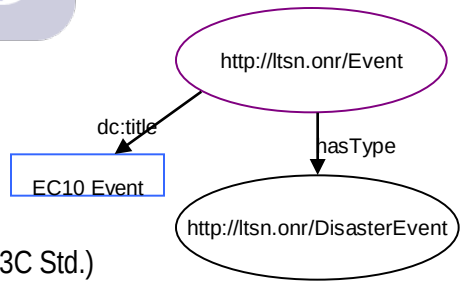


Approach: Characterize and Investigate

1. Observe actual composition and structure, and define nature of changes.
2. Investigate areas of theory that can accommodate observations.
3. Investigate initial feasibility.

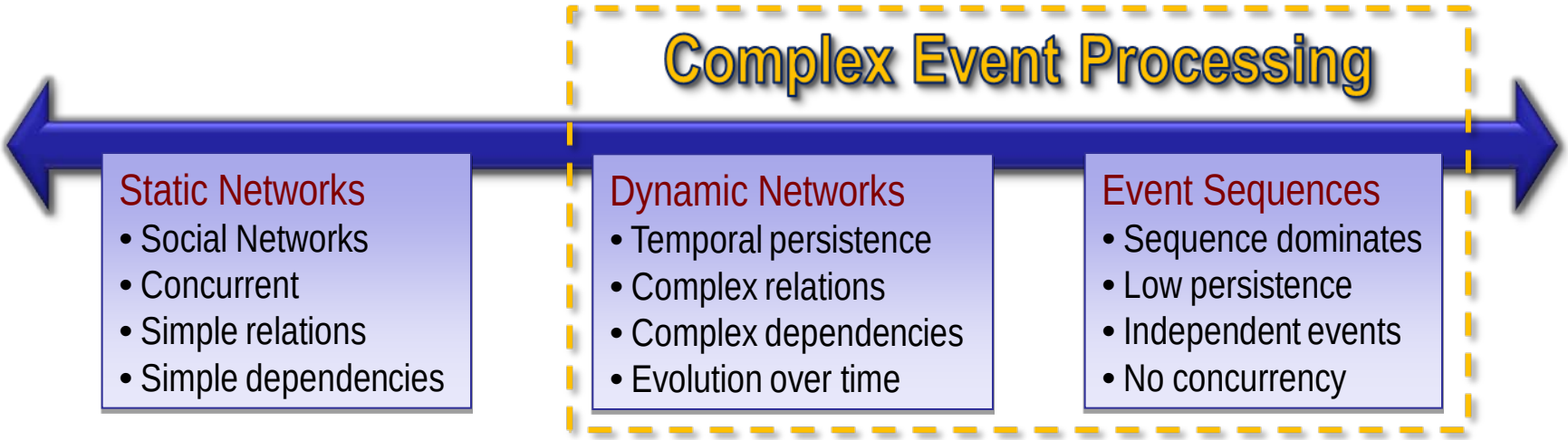
Data Environment





* RDF: Resource Description Format (W3C Std.)
http://www.w3.org/TR/rdf-primer/

The Event Continuum

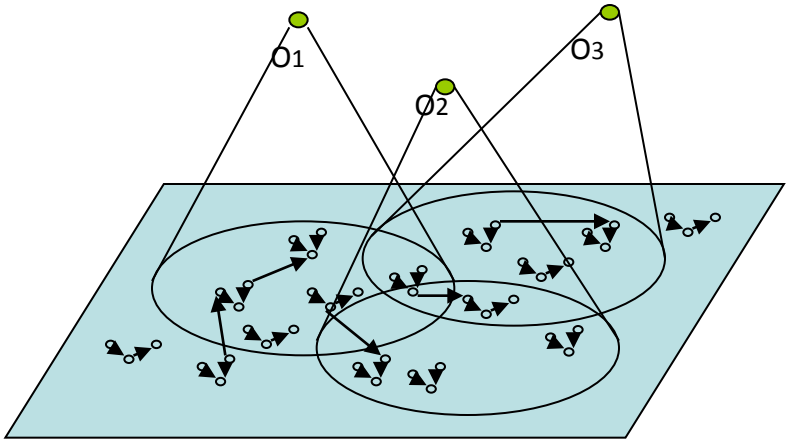


Three distinct regimes in the literature. Complex Event Processing in AIW event streams rarely has the characteristics of Static Networks.

Technical Approach

Characterization:

- Variable temporal persistence.
- Independent events.
- Complex dependences.
- Evolution.
- Sequential events.
- Concurrent events.



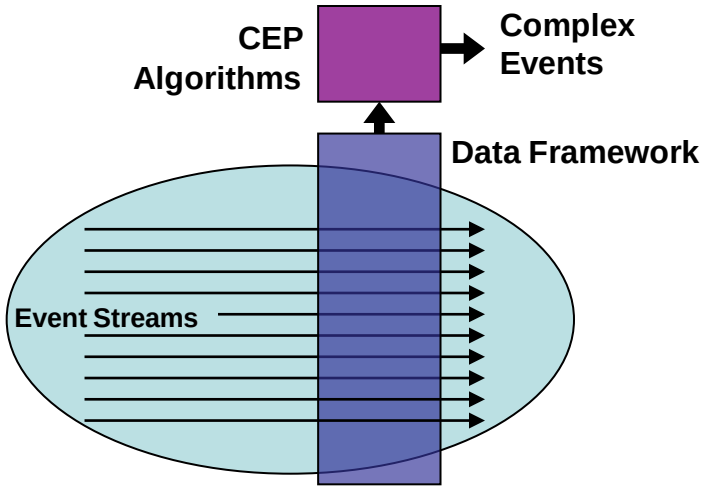
Investigate:

- Dynamic graph theory.

Initial feasibility:

- Algorithm prototype.

Prototype Implementation



A Framework and Algorithms to infer complex events from high-volume streams of events in near-real time.

Technical Approach

The approach involves the following steps:

1) Event Streams: Investigate structure and phenomenology of EIW events and event streams. Augment streams by adding context, time stamps, pedigree and graph structure.
2) Data Framework: Tag and encode the data using lexicons, schemas, and ontologies. Investigate dynamic graph theory towards scalable graph update and search.
3) CEP Algorithms: Develop and implement new dynamic graph algorithms for approximate graph pattern search. Evaluate algorithms to identify relevant patterns, activities, and events.
4) MOP Evaluation: establish means to measure and improve performance.

Simplified Search Algorithm

Graph Pattern Match

Input: Pattern P=(Vp,Ep), Data Graph G=(V,E), and Ontology O

Initial step: compute all-pairs-shortest path matrix M.

A. Find descendent nodes, D_{Vp}=desc(Vp), in ontology of each pattern graph node.

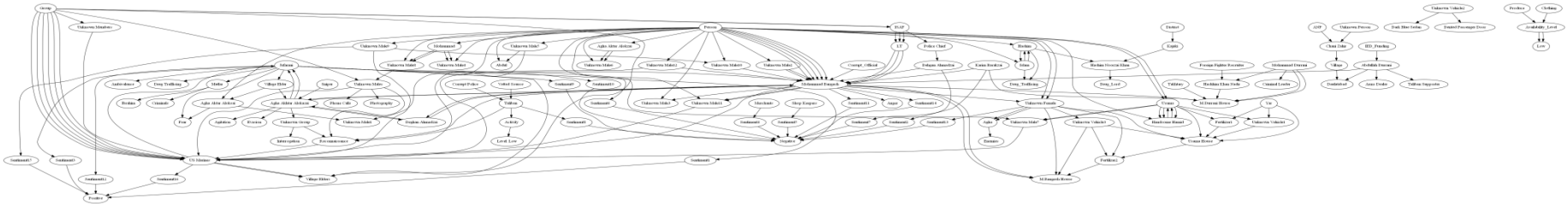
B. Compute potential matching set in G for each node in D_{Vp}.

C. Traverse paths in the matching set, examining path length and edge type. Remove nodes that are not connected, do not meet path constraints, or do not have correct edge type.

Distance matrix is updated as graph changes using algorithms from Ramalingam, 1996.

IIMEF Exercise: Key Leader Engagement (KLE)

KLE IPB Data Graph



Draft Intelligence Information Reports

DIIR_1-01.txt	DIIR_1-03.txt
DIIR_1-04.txt	DIIR_1-05.txt
DIIR_1-06.txt	DIIR_1-08.txt
DIIR_2-01.txt	DIIR_2-02.txt
DIIR_2-05.txt	DIIR_3-01.txt
DIIR_3-03.txt	DIIR_3-04.txt
DIIR_4-01.txt	DIIR_4-03.txt
DIIR_4-05.txt	DIIR_4-08.txt
DIIR_4-10.txt	DIIR_4-14.txt
DIIR_4-17.txt	

TACREPS

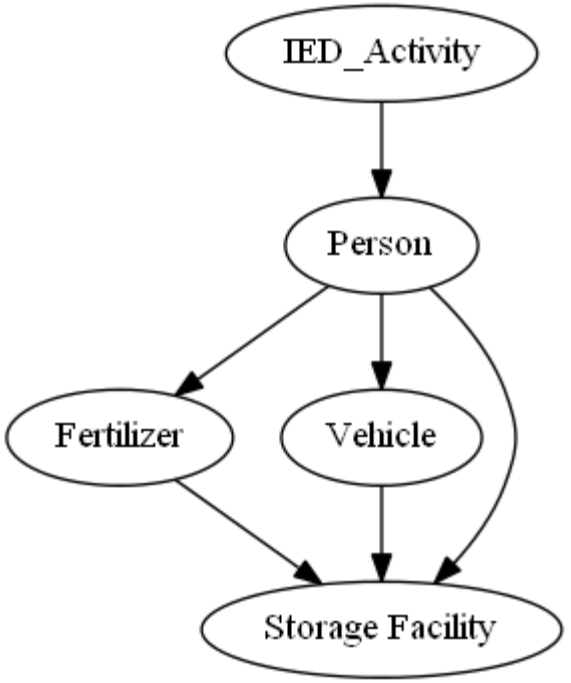
TACREP_1-02.txt	TACREP_2-03.txt
TACREP_2-04.txt	TACREP_2-06.txt
TACREP_3-02.txt	TACREP_4-02.txt
TACREP_4-04.txt	TACREP_4-06.txt
TACREP_4-09.txt	TACREP_4-11.txt
TACREP_4-12.txt	TACREP_4-13.txt
TACREP_4-16.txt	TACREP_4-18.txt
TACREP_4-19.txt	TACREP_4-21.txt

Dynamic Data Graph is built incrementally, as information becomes available.
Threat patterns can be detected at any time.

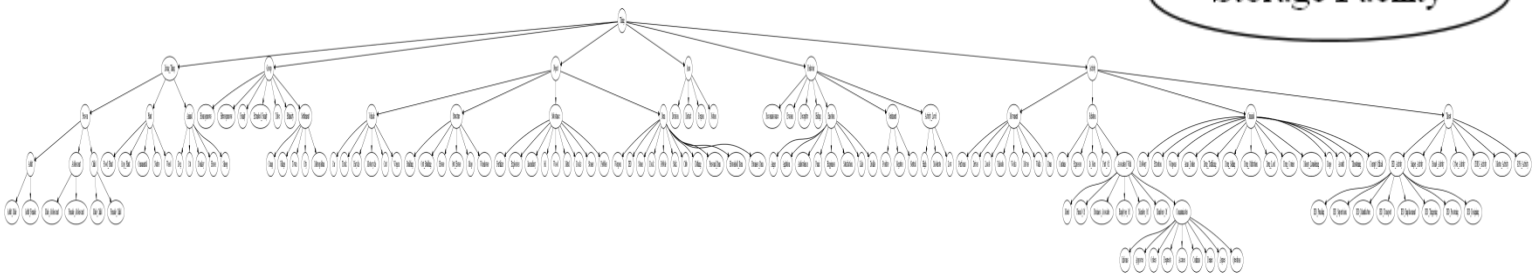
IPB: Intelligence Preparation for the Battlefield.

Threat Pattern Graph

Watch for a Person with a previous involvement in IED Activity, and access to Fertilizer, a Vehicle, and a Storage Facility.

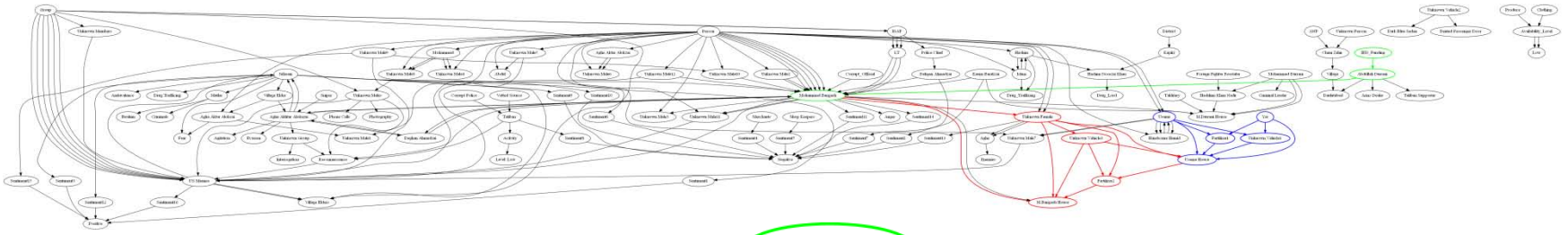


Given some hierarchy of objects and activities:



Watch the data as the graph changes and warn of any matching pattern.

Threat Warning Graph

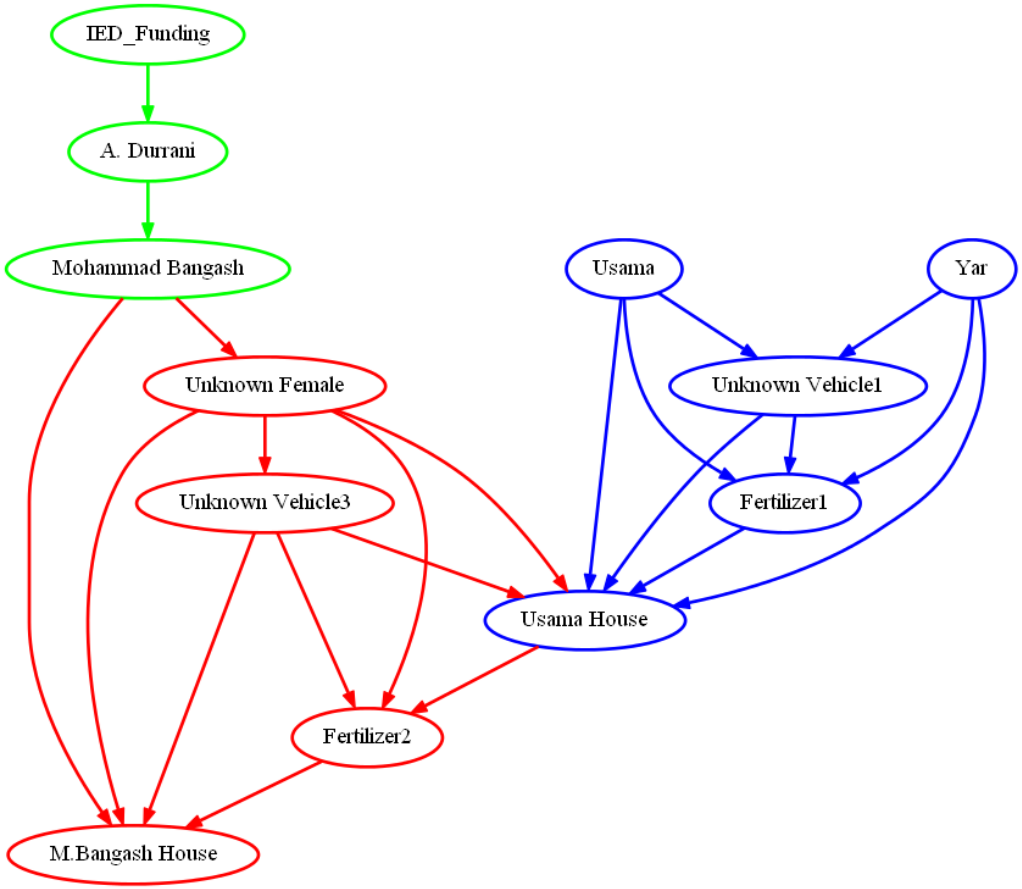


Draft Intelligence Information Reports

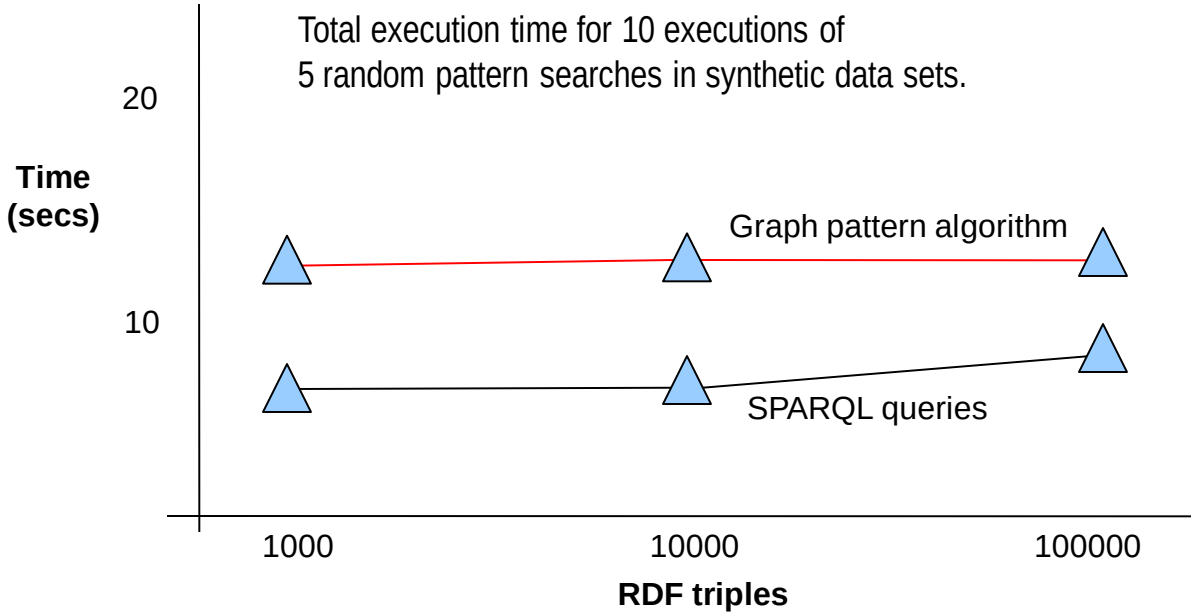
DIIR_1-01.txt	DIIR_1-03.txt
DIIR_1-04.txt	DIIR_1-05.txt
DIIR_1-06.txt	DIIR_1-08.txt
DIIR_2-01.txt	DIIR_2-02.txt
DIIR_2-05.txt	DIIR_3-01.txt
DIIR_3-03.txt	DIIR_3-04.txt
DIIR_4-01.txt	DIIR_4-03.txt
DIIR_4-05.txt	DIIR_4-08.txt
DIIR_4-10.txt	DIIR_4-14.txt
DIIR_4-17.txt	

TACREPS

TACREP_1-02.txt	TACREP_2-03.txt
TACREP_2-04.txt	TACREP_2-06.txt
TACREP_3-02.txt	TACREP_4-02.txt
TACREP_4-04.txt	TACREP_4-06.txt
TACREP_4-09.txt	TACREP_4-11.txt
TACREP_4-12.txt	TACREP_4-13.txt
TACREP_4-16.txt	TACREP_4-18.txt
TACREP_4-19.txt	TACREP_4-21.txt



Initial Performance Comparisons



Graph Pattern algorithm prototype:
- 680 line implementation in Python
- Uses networkx graph library.
- Uses rdflib RDF library.

SPARQL query prototype:
- 200 line implementation in Python
- Uses rdflib RDF library.
- Uses rdflib SPARQL library.

Execution environment: Dell Precision T1500, Core i7 processor, 8GB RAM, Windows 7 OS, 1TB HD.

Results

- Completed initial studies of EIW data streams and content.
- Completed initial prototype of graph pattern search algorithms.
- Completed initial performance comparisons with std. search algorithms *(Naïve implementation was no worse than current standard algs, with improved capabilities)*.
- Developed graph encoding framework, including activity and event hierarchy.
- Developed preliminary methods of performance assessment for activity discovery.
- Implemented prototype incremental graph.

Issues

Data issues:

- Access to suitable data with documentation.
- Relatively small data set sizes.
- Time constraints for processing and encoding data.

Future Work

- Mature graph encoding work for wider application and greater robustness.
- Automate conditioning of data to facilitate larger-scale testing.
- Participate in user experiments, when/where possible.
- Investigate additional application areas with larger data sets.